# LEARNING PROTEIN SEQUENCE EMBEDDINGS USING INFORMATION FROM STRUCTURE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Inferring structural properties of a protein given only its amino acid sequence is a challenging problem. Existing approaches based solely on sequence are unable to recognize and exploit structural patterns when sequences have diverged too far. We introduce a novel framework for infusing structural information into position-specific representations of protein sequences. We train bidirectional long short-term memory (LSTM) models on protein sequences with a two-part feedback mechanism to incorporate (i) pairwise residue contact maps for individual proteins and (ii) co-membership in structural categories based on a curated database (SCOPe). For co-membership, we introduce soft symmetric alignment (SSA) between sequences of vector embeddings. We show empirically that our approach outperforms existing direct sequence alignment methods and also a structure-based alignment method when predicting structural similarity. SSA also enables learning informative position-specific embeddings even when no residue level supervision is available. Finally, we demonstrate that the learned embeddings can be transferred to other protein sequence problems, improving state-of-the-art in transmembrane domain prediction. [1]

## 1 INTRODUCTION

Proteins are linear chains of amino acids that fold into specific 3d conformations as a result of the physical properties of the amino acid sequence. These structures, in turn, determine the wide array of functions carried out by proteins, from binding specificity to catalytic activity to localization within the cell. Information about structure is vital for studying the mechanisms of these molecular machines in health and disease, and for development of new therapeutics. However, structures have only been experimentally measured for a tiny fraction of known proteins. Thus, we seek to transfer information from proteins with known structure to those with similar but unknown structure. However, this is a challenging problem, because primary sequence similarity only loosely relates to structural similarity [1, 2, 3, 4].

In natural language processing, representation learning has proven effective for knowledge transfer between tasks. Word embeddings [5, 6, 7], and more recently, sentence embeddings [8, 9, 10, 11], have been shown to be invaluable as features for problems ranging from machine translation to natural language inference. In the same way that sentences consist of a sequence of words that encode semantic meaning, proteins consist of a linear sequence of amino acids that fold into a 3D structure and determine function. Despite this similarity, methods for learning protein sequence embedding models remain largely unexplored. Existing works focus on unsupervised approaches based on k-mer co-occurrence and hence do not capture information from known protein structures [12, 13].

In this work, we address the problem of learning protein sequence embeddings using weak supervision from global structural similarity. Given only observations of the structural similarity between pairs of sequences, the task is to learn a contextual embedding model such that residues occurring in similar local structures will be close in embedding space without access to direct observations of position-level correspondences between sequences. We solve this problem by explicitly decomposing sequence comparison into an alignment of the sequences and a pairwise comparison of the aligned residues in embedding space. We introduce a soft symmetric alignment (SSA) mechanism — a

[1] source code and datasets are available at **redacted**

symmetrization of the directional alignment commonly used in attention mechanisms — and use this alignment mechanism to train bidirectional recurrent neural network encoders to give residue embeddings. Furthermore, in order to take advantage of information about local structural context within proteins, we extend this framework to include supervision from residue-residue contacts in the individual protein structures (Figure 1). This multitask framework allows us to newly leverage both global structural similarity between proteins and residue-residue contacts within proteins for training embedding models.

We apply this framework to the SCOPe ASTRAL dataset (version 2.06) [14], containing structures for 28,010 proteins with maximum sequence identity of 95%. SCOPe manually classifies protein structures into a hierarchy of class, fold, superfamily, and family (Appendix Figure 3), where co-membership corresponds to increasing structural similarity. On this dataset, a bidirectional LSTM encoder trained with our multitask framework dramatically outperforms sequence homology-based protein comparison methods and even outperforms the widely used TMalign structural alignment tool [15] despite using no structural information at test time. Furthermore, we demonstrate that our SSA mechanism outperforms alternative alignment methods for predicting structural similarity from sequence embeddings and that inclusion of the contact prediction training task further improves structural similarity prediction and positional embedding quality. Finally, we demonstrate that the embeddings learned by our model are generally applicable to other protein machine learning problems by leveraging our embeddings to improve the state-of-the-art in transmembrane prediction. This work presents the first attempt in learning protein sequence embeddings from structure and takes a step towards bridging the sequence-structure divide with representation learning.

## 2 Related Work

Current work in protein sequence embeddings has seen little adoption and been restricted largely to adaptations of unsupervised methods from NLP such as ProtVec [12] and Yang et al. [13]. Other methods have focused on manual feature engineering based on biophysical and sequence attributes [16]. Pretrained word embeddings, derived from large unlabeled text corpora, [6, 7] are widely used in most NLP architectures but only contain context-independent information. Sentence and document embedding methods focus on learning single vector representations of whole sequences using mostly unsupervised approaches [17, 18, 8, 9]. Recently, supervised sentence encoders trained on natural language inference have also shown promising results [19, 20, 11]. An alternative line of work seeks to learn context-dependent representations using bidirectional recurrent neural network language models [21, 22] or encoders trained for machine translation [23]. We show that a bidirectional RNN language model similar to [21, 22] is useful for improving performance of our structural similarity models, the first application of this approach to biological sequences. Our primary contribution, however, focuses instead on learning context embedding models in a supervised sequence comparative framework relevant to protein structural embedding.

This work is also related to approaches for learning cross lingual word embeddings from unaligned parallel text. Kočiský et al. [24] learn bilingual word embeddings jointly with a FastAlign [25] word alignment model using expectation maximization. BilBOWA [26] learns cross lingual word embeddings using parallel sentences without word level alignments by assuming a uniform alignment between words. In contrast to these approaches, we are interested in learning an RNN context embedding model and alignment model jointly in order to capture higher order sequence semantics. Word Mover's Distance [27], in which documents are compared based on an alignment between their words using pretrained word vectors, and it's supervised variant [28], in which word vector's are linearly transformed given document class supervision, are also related. However, the former focuses only on measuring document similarity and does not learn embeddings whereas the latter learns a linear transformation of the word embeddings in a nearest-neighbor framework. Furthermore, the WMD alignments are expensive to compute scaling as $O(p^3 \log p)$ where $p$ is the number of unique words. This is prohibitive when alignments must be computed at every optimization step. In contrast, our soft symmetric alignment method only scales as $O(nm)$ where $n$ and $m$ are the lengths of the sequences being aligned.

Our SSA method is partly inspired by previous work using soft alignments and attention mechanisms for sequence modeling. Bahdanau et al. [29] and Hermann et al. [30] use dynamic attention mechanisms to form directional soft alignments for encoder-decoder models. Our approach is both

memoryless and symmetric. Several additional works use query-to-document and document-to-query attention mechanisms for reading comprehension [31, 32, 33] but do not combine these into a single symmetric alignment. Furthermore, the pairwise word interaction model [34] uses a "hard" alignment upweighting with 19-layer CNN aggregation scheme and the decomposable attention model [35] and enhanced sequential inference model [35] explore some additional asymmetric alignment ideas with elaborate aggregation schemes for natural language inference. However, none of the above methods are focused on learning useful per element embeddings.

Protein fold recognition is the problem of classifying proteins into folds (for example, as defined by the SCOP database) based on their sequences. Approaches to this problem have largely been based on sequence homology using sequence similarity to classify structures based on close sequence matches [14, 36]. These methods are either direct sequence alignment tools [37, 38] or based on profile HMMs in which multiple sequence alignments are first built by iterative search against a large sequence database, the multiple sequence alignments are converted into profile HMMs, and then sequences are compared using HMM-sequence or HMM-HMM alignments [39, 40]. However, these methods are only appropriate for matching proteins with high sequence similarity [2, 3, 36]. In contrast, we focus on learning protein sequence representations that capture structure information in an easily transferable manner directly from structural comparison supervision. We hypothesize that this method will improve structural prediction for proteins with low sequence homology.

## 3 METHODS

Our model is composed of three main components: (1) a 3-layer bidirectional LSTM encoder that transforms amino acid sequences into sequences of vector embeddings with additional inputs from a pretrained LSTM language model, (2) structurally similarity prediction using the SSA mechanism to align sequences based on the vector embeddings, and (3) contact prediction using a convolutional layer on top of vectors given by the pairwise comparison of positions within the protein sequences.

### 3.1 SEQUENCE ENCODER WITH PRETRAINED LANGUAGE MODEL

The encoder takes a sequence of amino acids representing a protein and encodes it into a sequence of vector representations of the same length. We structure the encoder as a stack of bidirectional LSTMs followed by a linear layer projecting the outputs of the final LSTM layer into the output embedding space (Appendix Figure 2). In order to transfer knowledge from all of known protein space, we include the hidden layer outputs of a bidirectional language model pretrained on the raw protein sequences in the protein families database (Pfam) [41] as inputs to the encoder with a learned linear transformation as

$$h_i^{input} = \text{ReLU}(W^{LM} h_i^{LM} + W^x x_i + b)$$

where $h_i^{LM}$ is the concatenation of the LM hidden states at position $i$, $x_i$ is a one-hot encoding of the amino acid at position $i$, and $W^{LM}$, $W^x$, and $b$ are learned parameters. Throughout this work, the parameters of the language model are frozen. Only $W^{LM}$, $W^x$, and $b$ are updated when learning the encoder. Training details for the language model can be found in the Appendix section A.1.

### 3.2 PROTEIN STRUCTURE COMPARISON

We define the structural similarity between two proteins based on their SCOP class as the number of levels shared by those proteins in the SCOP hierarchy (Appendix Figure 3). For example, proteins sharing the same class but not the same fold would have $y = 1$, while two proteins sharing the same superfamily but not the same family would have $y = 3$. Proteins not sharing the same class are assigned $y = 0$.

#### 3.2.1 SOFT SYMMETRIC ALIGNMENT

In order to calculate the similarity of two amino acid sequences given that each has been encoded into a sequence of vector representations, $z_1 ... z_n$ and $z'_1 ... z'_m$, we develop a soft symmetric alignment mechanism in which the similarity between two sequences is calculated based on their vector embeddings as

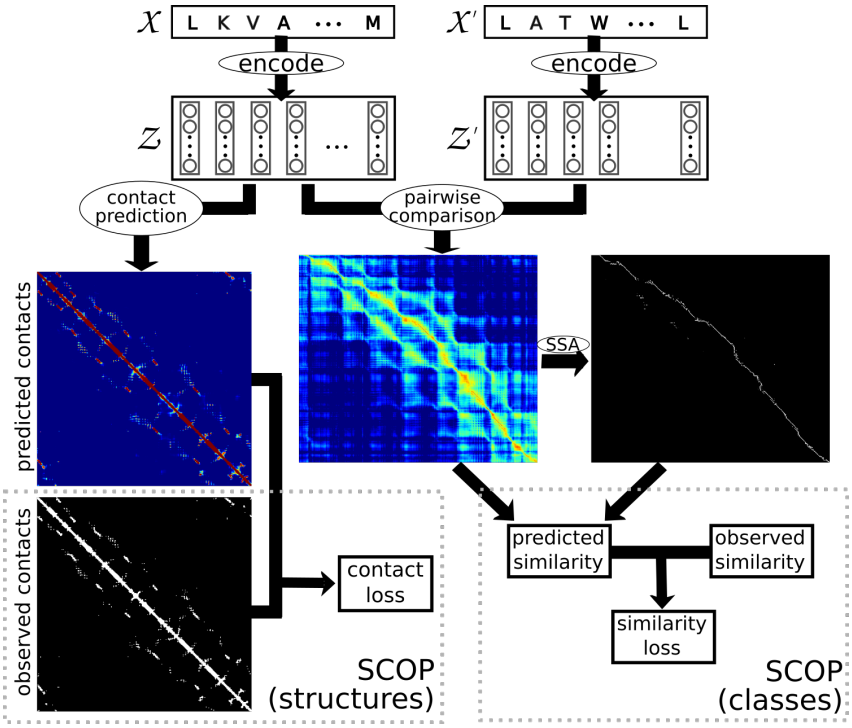


Figure 1: Diagram of the various components of the loss function. Vector embeddings from the encoder are used to predict contacts between residues within a single sequence and to predict similarity between sequences using SSA. Observed contact maps and structural similarities from SCOP are used to train the model. The full loss is given by a weighted sum of the similarity and contact losses.

$$\hat{s} = -\frac{1}{A}\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}||z_i - z'_j||_1$$

where $a_{ij}$ are entries of the alignment matrix given by

$$\alpha_{ij} = \frac{\exp(-||z_i - z'_j||_1)}{\sum_{k=1}^{n}\exp(-||z_i - z'_k||_1)} \quad \text{and} \quad \beta_{ij} = \frac{\exp(-||z_i - z'_j||_1)}{\sum_{k=1}^{m}\exp(-||z_k - z'_j||_1)}$$
$$a_{ij} = \alpha_{ij} + \beta_{ij} - \alpha_{ij}\beta_{ij}$$

and $A = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}$ is the length of the alignment.

### 3.2.2 Ordinal regression

Next, to relate this scalar similarity to the ordinal structural similarities, we adopt an ordinal regression framework in which we jointly learn a set of binary linear classifiers with parameters $\theta_1 ... \theta_4$ and $b_1 ... b_4$ to predict whether the structural similarity is greater than or equal to $t$, $\hat{p}(y \geq t) = \text{sigmoid}(\theta_t \hat{s} + b_t)$, with the constraint that $\theta_t \geq 0$ to enforce that $\hat{p}$ increases monotonically with $\hat{s}$. The structural similarity loss is then given by

$$\mathop{\mathbb{E}}_{x,x'}\left[\sum_{t=1}^{4}(y \geq t)\log(\hat{p}(y \geq t)) + (y < t)\log(1 - \hat{p}(y \geq t))\right]$$

Given these classifiers, the predicted probability that two sequences share structural similarity at level $t$ is $\hat{p}(y = t) = \hat{p}(y \geq t)(1 - p(y \geq t + 1))$ with $\hat{p}(y \geq 0) = 1$ by definition.

### 3.3 RESIDUE-RESIDUE CONTACT PREDICTION

In addition to the structural similarity supervision, we also leverage within sequence information in the form of residue-residue contacts for learning embedding models. Contact prediction is a binary classification problem in which we seek to predict whether residues at positions $i$ and $j$ within a protein make contact in the 3d structure. In this work, we define contacts as being residues with $C\alpha$ atoms within 8Åof each other. In order to predict contacts from the sequence of embedding vectors given by the encoder, we define a tensor of size (NxNx2D) where D is the dimension of the embedding vector containing pairwise features given by the concatenation of the absolute element-wise differences and the element-wise products of the embedding vectors for each pair of positions, $v_{ij} = [|z_i - z_j|; z_i \odot z_j]$. These vectors are then transformed through a single hidden layer with 50 units (implemented as a width 1 convolutional layer) and ReLU activation as in $h_{ij} = \text{ReLU}(W v_{ij} + b)$. Contact predictions are then given by the sigmoid of a 7x7 convolutional filter with a single output channel applied to the NxNx50 tensor given by the previous operation. The parameters of the hidden layer and convolutional filter are learned jointly with the encoder to minimize the cross entropy between the predicted and true contacts, $L^{contact}$, over all pairs of residues within all proteins in the training set. The combined loss function is then given by

$$\lambda L^{similarity} + (1 - \lambda) L^{contact}$$

where $\lambda$ is a parameter that interpolates between the structural similarity and contact prediction losses. Throughout this work, we refer to models this full framework as "SSA with contact prediction" and the framework without contact prediction ($\lambda = 1$) as "SSA" or "SSA without contact prediction."

### 3.4 TRAINING DETAILS

In all experiments, models were trained for 100 epochs using ADAM with a learning rate of 0.001. Each epoch consisted of 100,000 examples sampled from the SCOP structural similarity training set with smoothing of the similarity level distribution of 0.5. In other words, the probability of sampling a pair of sequences with similarity level $t$ is proportional to $N_t^{0.5}$ where $N_t$ is the number of sequence pairs with $t$ similarity in the training set. The structural similarity component of the loss is estimated with minibatches of 64 pairs. When using the full multitask objective, the contact prediction component uses minibatches of 10 sequences and $\lambda = 0.1$. Furthermore, during training we apply a small perturbation to the sequences by resampling the amino acid at each position from the uniform distribution with probability 0.05. All models were implemented in pytorch and experiments were run on a single NVIDIA Tesla V100 GPU.

## 4 RESULTS

### 4.1 SCOP STRUCTURAL RELATEDNESS

We first consider the structure similarity prediction problem using the SCOPe ASTRAL 2.06 dataset filtered to 95% sequence identity. We split this dataset into 22,408 train and 5,602 heldout sequences. From the heldout sequences, we randomly sample 100,000 pairs as the structural similarity test set. Furthermore, we define a second test set using the newest release of SCOPe (2.07) by collecting all protein sequences added between the 2.06 and 2.07 ASTRAL releases. This gives a set of 688 protein sequences all pairs of which define the ASTRAL 2.07 new test set. We compare the performance of our full model against a number of widely-used protein sequence comparison methods and a structural-alignment based method (see Appendix section A.2). We find that our embedding model with SSA comparison outperforms all sequence alignment based methods by a large margin. We improve overall prediction accuracy from 0.79 to 0.95, Pearson's correlation from 0.37 to 0.91, and Spearman's rank correlation from 0.23 to 0.69 over the next best sequence comparison method, HHalign [40]. We also consider the average-precision score for predicting pairs of proteins at each level of similarity (i.e. positives are considered all pairs with class or higher similarity, fold or higher similarity, superfamily or higher similarity, or family level similarity) and find that our embedding model improves over HHalign by 0.51, 0.28, 0.09, and 0.12 at each level on the 2.06 test. Our embedding model also improves on all metrics on the new 2.07 proteins test set. We note that

| | | Correlation | | Average-precision score | | | |
|---|---|---|---|---|---|---|---|
| **Model** | **Accuracy** | $r$ | $\rho$ | **Class** | **Fold** | **Superfamily** | **Family** |
| ASTRAL 2.06 test set | | | | | | | |
| NW-align | 0.78462 | 0.18854 | 0.14046 | 0.30898 | 0.40875 | 0.58435 | 0.52703 |
| phmmer [HMMER 3.2.1] | 0.78454 | 0.21657 | 0.06857 | 0.26022 | 0.34655 | 0.53576 | 0.50316 |
| HHalign [HHsuite 3.0.0] | 0.78851 | 0.36759 | 0.23240 | 0.40347 | 0.62065 | 0.86444 | 0.52220 |
| TMalign | 0.80831 | 0.61687 | 0.37405 | 0.54866 | 0.85072 | 0.83340 | 0.57059 |
| SSA w/ contact prediction | **0.95149** | **0.90954** | **0.69018** | **0.91458** | **0.90229** | **0.95262** | **0.64781** |
| ASTRAL 2.07 new test set | | | | | | | |
| NW-align | 0.80842 | 0.37671 | 0.23101 | 0.43953 | 0.77081 | 0.86631 | 0.82442 |
| phmmer [HMMER 3.2.1] | 0.80907 | 0.65326 | 0.25063 | 0.38253 | 0.72475 | 0.82879 | 0.81116 |
| HHalign [HHsuite 3.0.0] | 0.80883 | 0.68831 | 0.27032 | 0.47761 | 0.83886 | 0.94122 | 0.82284 |
| TMalign | 0.81275 | 0.81354 | 0.39702 | 0.59277 | 0.91588 | 0.93936 | 0.82301 |
| SSA w/ contact prediction | **0.93151** | **0.92900** | **0.66860** | **0.89444** | **0.93966** | **0.96266** | **0.86602** |

Table 1: Comparison of the full SSA w/ contact prediction embedding model with three protein sequence alignment methods (NW-align, phmmer, and HHalign) and the structure alignment method TMalign. We measure accuracy, Pearson's correlation ($r$), Spearman's rank correlation ($\rho$), and average-precision scores for retrieving protein pairs with structural similarity of at least class, fold, superfamily, and family levels.

average-precision scores are higher for fold, superfamily, and family levels of this dataset due to sequences being more concentrated in a small number of families than in the 2.06 test set.

For reference, we also compare with TMalign, a method for computing structural similarity based on alignment of actual 3d protein structures [15]. Given the 3d structures of query and target proteins, TMalign reports scores for query to target and target to query structural alignments. We use the average of the scores for each direction to calculate the TMalign predictions. Remarkably, despite using only amino acid sequences as input, our embedding model even outperforms the TMalign *structure* alignment method on all metrics for both datasets. The largest improvement comes at the class level where TMalign achieves much lower average-precision score for the retrieving the weak structural matches than do our embeddings.

### 4.2 Evaluation of model components

We next evaluate the individual model components on two tasks: structure similarity prediction on the ASTRAL 2.06 test set and 8-class secondary structure prediction on a 40% sequence identity filtered dataset containing 22,086 protein sequences from the protein data bank (PDB) [42], a repository of experimentally determined protein structures. Secondary structure prediction is a sequence labeling problem in which we attempt to classify every position of a protein sequence into one of eight classes describing the local 3d structure at that residue. We use this task to measure the utility of our embeddings for position specific prediction problems. For this problem, we split the secondary structure dataset into 15,461 training and 6,625 testing sequences. We then treat each position of each sequence as an independent datapoint with features either given by the 100-d embedding vector or 1-hot encoding of the k-mer at that position and train a fully connected neural network (2 hidden layers, 1024 units each, ReLU activations) to predict the secondary structure class from the feature vector. These models are trained with cross entropy loss for 10 epochs using ADAM with learning rate 0.001 and a minibatch size of 256.

**Alignment from vector embeddings.** We first demonstrate the importance of our SSA mechanism when training the contextual embedding model by comparing the performance of models trained with SSA versus models trained with uniform alignment and a mean embedding comparison approaches [26]. In uniform alignment (UA), we consider a uniform prior over possible alignments giving the similarity score. For the mean embedding method (ME), we instead calculate the similarity score based on the difference between the average embedding of each sequence.

| | Structural similarity | | | Secondary structure | |
|---|---|---|---|---|---|
| **Embedding Model/Features** | **Accuracy** | $r$ | $\rho$ | **Perplexity** | **Accuracy** |
| 1-mer | - | - | - | 4.804 | 0.374 |
| 3-mer | - | - | - | 4.222 | 0.444 |
| 5-mer | - | - | - | 5.154 | 0.408 |
| SSA w/o language model | 0.89847 | 0.81459 | 0.64693 | 3.818 | 0.511 |
| ME | 0.92821 | 0.85977 | 0.67122 | 4.058 | 0.480 |
| UA | 0.93524 | 0.87536 | 0.67017 | 4.470 | 0.427 |
| SSA | 0.93794 | 0.88048 | 0.67645 | 4.027 | 0.487 |
| SSA w/ contact prediction | **0.95149** | **0.90954** | **0.69018** | **2.861** | **0.630** |

Table 2: Study of individual model components. Results of structural similarity prediction on the ASTRAL 2.06 test set and secondary structure prediction are provided for embedding models trained with various alignment methods, the SSA embedding model trained with language model inputs, and for the full SSA embedding model trained with contact prediction.

$$\hat{s}^{UA} = -\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}||z_i - z'_j||_1 \qquad \text{and} \qquad \hat{s}^{ME} = -\left|\left|\frac{1}{n}\sum_{i=1}^{n} z_i - \frac{1}{m}\sum_{i=1}^{m} z'_j\right|\right|_1$$

We find that not only are the SSA embeddings better predictors of secondary structure than k-mer features (accuracy 0.487 vs. 0.444 for 3-mers), but that the SSA mechanism is necessary for achieving best performance on both the structure similarity and local structure prediction tasks. As seen in table 2, the ME model achieves close to SSA performance on secondary structure prediction, but is significantly worse for SCOP similarity prediction. The UA model, on the other hand, is close to SSA on SCOP similarity but much worse when predicting secondary structure. This suggests that our SSA mechanism captures the best of both methods, allowing embeddings to be position specific as in the ME model but also being better predictors of SCOP similarity as in the UA model.

**Within sequence contact prediction.** Although the SSA mechanism allows our embedding model to capture position specific information, we wanted to explore whether positional information *within* sequences in the form of contact prediction could be used to improve the embeddings. We train models with and without the contact prediction task and find that including contact prediction improves both the structural similarity prediction and secondary structure prediction results. The accuracy of secondary structure prediction improves from 0.487 without to 0.630 with contact prediction (Table 2). This suggests that the contact prediction task dramatically improves the quality of the local embeddings on top of the weak supervision provided by whole structure comparison.

**Pretrained language model.** Finally, we show, for the first time with biological sequences, that a language model pretrained on a large unsupervised protein sequence database can be used to transfer information to supervised sequence modeling problems. SCOP similarity classification results for SSA embedding models trained with and without LM hidden layer inputs shows that including the LM substantially improves performance, increasing accuracy from 0.898 to 0.938 simply by including representations learned on the Pfam database. Furthermore, we examine the extent to which LM hidden states capture all useful structural information by training SSA embedding models with less expressive power than our 3-layer LSTM architecture (Appendix Table 4). We find that the LM hidden states are not sufficient for high performance on the structural similarity task with linear and fully connected (width 1 convolution) embedding models only achieving 0.853 and 0.910 accuracy on the ASTRAL 2.06 test set respectively.

### 4.3 TRANSMEMBRANE PREDICTION

We demonstrate the potential utility of our protein sequence embedding model for transfering structural information to other sequence prediction problems by leveraging our embedding model

| | Prediction category | | | | |
|---|---|---|---|---|---|
| **Method** | **TM** | **SP+TM** | **Globular** | **Globular+SP** | **Overall** |
| TOPCONS | **0.80** | 0.80 | 0.97 | 0.91 | 0.87 |
| MEMSAT-SVM | 0.67 | 0.52 | 0.88 | 0.00 | 0.52 |
| Philius | 0.70 | 0.75 | 0.94 | 0.94 | 0.83 |
| Phobius | 0.55 | 0.83 | 0.95 | 0.94 | 0.82 |
| PolyPhobius | 0.55 | 0.83 | 0.95 | 0.94 | 0.82 |
| SPOCTOPUS | 0.71 | 0.78 | 0.78 | 0.79 | 0.76 |
| CRF (SSA w/ contact prediction) | 0.77 | **0.85** | **1.00** | 0.94 | **0.89** |

Table 3: Accuracy of transmembrane prediction using structural embeddings in 10-fold cross validation and comparison with other transmembrane prediction methods. Our CRF with potentials given by a function of the SSA with contact prediction embeddings results are displayed below the dotted line. We compare with results for a variety of transmembrane prediction methods previously reported on the TOPCONS dataset.

for transmembrane prediction. In transmembrane prediction, we wish to detect which, if any, segments of the amino acid sequence cross the lipid bilayer for proteins integrated into the cell membrane. This is a well studied problem in protein biology with methods generally consisting of HMMs with sophisticated, manually designed hidden state transition distributions and emission distributions including information about residue identity, amino acid frequencies from multiple sequence alignments, local structure, and chemical properties. Newer methods are also interested in detection of signal peptides, which are short amino acid stretches at the beginning of a protein sequence signaling for this protein to be inserted into the cell membrane.

To benchmark our embedding vectors for this problem, we develop a conditional random field model in which propensity of each hidden state given the sequence of embedding vector inputs is defined by a single layer bidirectional LSTM with 100 units. For the transition probabilities between states, we adopt the same structure as used in TOPCONS [43] and perform 10-fold cross validation on the TOPCONS transmembrane benchmark dataset. We report results for correctly predicting regions in proteins with only transmembrane domains (TM), transmembrane domains and a signal peptide (SP+TM), neither transmembrane nor signal peptide domains (Globular), or a signal peptide but no transmembrane regions (Globular+SP). Transmembrane state labels are predicted with Viterbi decoding. Again following TOPCONS, predictions are counted as correct if, for TM proteins, our model predicts no signal peptide, the same number of transmembrane regions, and those regions overlap with real regions by at least five positions. Correct SP+TM predictions are defined in the same way except that proteins must be predicted to start with a signal peptide. Globular protein predictions are correct if no transmembrane or signal peptides are predicted and Globular+SP predictions are correct if only a leading signal peptide is predicted.

We find that our transmembrane predictions rank first or tied for first in 3 out of the 4 categories (SP+TM, Globular, and Globular+SP) and ranks second for the TM category. Overall, our transmembrane predictions are best with prediction accuracy of 0.89 vs 0.87 for TOPCONS. Remarkably, this is by simply replacing the potential function in the CRF with a function of our embedding vectors, the hidden state grammar is the same as that of TOPCONS. This is particularly noteworthy, because TOPCONS is a meta-predictor. It uses outputs from a wide variety of other transmembrane prediction methods to define the transmembrane state potentials.

## 5 Conclusion

In this work, we proposed a novel alignment approach to learning contextual sequence embeddings with weak supervision from a global similarity measure. Our SSA model is fully differentiable, fast to compute, and can be augmented with within sequence structural information, outperforming competition in predicting protein structural similarity. The resulting embeddings are widely useful, allowing us to improve over the state-of-the-art in transmembrane region prediction. The broader framework extends to other related (non-biological) tasks.

## REFERENCES


[1] A G Murzin, S E Brenner, T Hubbard, and C Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 247(4):536–540, April 1995.

[2] L Holm and C Sander. Mapping the protein universe. *Science*, 273(5275):595–603, August 1996.

[3] Hin Hark Gan, Rebecca A Perlow, Sharmili Roy, Joy Ko, Min Wu, Jing Huang, Shixiang Yan, Angelo Nicoletta, Jonathan Vafai, Ding Sun, Lihua Wang, Joyce E Noah, Samuela Pasquali, and Tamar Schlick. Analysis of protein sequence/structure similarity relationships. *Biophys. J.*, 83(5):2781–2791, November 2002.

[4] S E Brenner, C Chothia, T J Hubbard, and A G Murzin. Understanding protein structure: using scop for fold interpretation. *Methods Enzymol.*, 266:635–643, 1996.

[5] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(Feb):1137–1155, 2003.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[7] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[8] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-Thought vectors. In *Advances in Neural Information Processing Systems*, pages 3294–3302, 2015.

[9] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*, 2016.

[10] J Mueller and A Thyagarajan. Siamese recurrent architectures for learning sentence similarity. *AAAI*, 2016.

[11] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017.

[12] Ehsaneddin Asgari and Mohammad R. K. Mofrad. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLOS ONE*, 10(11):1–15, 11 2015. doi: 10.1371/journal.pone.0141287. URL `https://doi.org/10.1371/journal.pone.0141287`.

[13] K. K. Yang, Z. Wu, C. N. Bedbrook, and F. H. Arnold. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, Aug 2018.

[14] Naomi K Fox, Steven E Brenner, and John-Marc Chandonia. SCOPe: Structural classification of proteins–extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res.*, 42 (Database issue):D304–9, January 2014.

[15] Y. Zhang and J. Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.*, 33(7):2302–2309, 2005.

[16] D. Ofer and M. Linial. ProFET: Feature engineering captures high-level protein functions. *Bioinformatics*, 31(21):3429–3436, Nov 2015.

[17] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In Eric P Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China, 2014. PMLR.

[18] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*, 2017.

[19] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[20] Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. How transferable are neural networks in NLP applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489, 2016.

[21] Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1756–1765, 2017.

[22] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237, 2018.

[23] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305, 2017.

[24] Tomáš Kočiský, Karl Moritz Hermann, and Phil Blunsom. Learning bilingual word representations by marginalizing alignments. In *Proceedings of ACL*, pages 224–229, 2014.

[25] Chris Dyer, Victor Chahuneau, and Noah A Smith. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of NAACL-HLT*, pages 644–649, 2013.

[26] Stephan Gouws, Yoshua Bengio, and Greg Corrado. BilBOWA: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 748–756, Lille, France, 2015. PMLR.

[27] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

[28] Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha, and Kilian Q Weinberger. Supervised word mover's distance. In *Advances in Neural Information Processing Systems*, pages 4862–4870, 2016.

[29] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

[30] Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 1693–1701, Cambridge, MA, USA, 2015. MIT Press.

[31] Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 908–918, 2016.

[32] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 593–602, 2017.

[33] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*, 2017.

[34] Hua He and Jimmy Lin. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948, 2016.

[35] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*, 2016.

[36] John-Marc Chandonia, Naomi K Fox, and Steven E Brenner. SCOPe: Manual curation and artifact removal in the structural classification of proteins - extended database. *J. Mol. Biol.*, 429(3):348–355, February 2017.

[37] S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–410, October 1990.

[38] S B Needleman and C D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3):443–453, March 1970.

[39] R. D. Finn, J. Clements, and S. R. Eddy. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res.*, 39(Web Server issue):29–37, Jul 2011.

[40] J. Soding, A. Biegert, and A. N. Lupas. The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Res.*, 33(Web Server issue):W244–248, Jul 2005.

[41] R. D. Finn, A. Bateman, J. Clements, P. Coggill, R. Y. Eberhardt, S. R. Eddy, A. Heger, K. Hetherington, L. Holm, J. Mistry, E. L. Sonnhammer, J. Tate, and M. Punta. Pfam: the protein families database. *Nucleic Acids Res.*, 42(Database issue):D222–230, Jan 2014.

[42] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000. doi: 10.1093/nar/28.1.235. URL http://dx.doi.org/10.1093/nar/28.1.235.

[43] K. D. Tsirigos, C. Peters, N. Shu, L. Kall, and A. Elofsson. The TOPCONS web server for consensus prediction of membrane protein topology and signal peptides. *Nucleic Acids Res.*, 43(W1):W401–407, Jul 2015.

[44] T J Hubbard, A G Murzin, S E Brenner, and C Chothia. SCOP: a structural classification of proteins database. *Nucleic Acids Res.*, 25(1):236–239, January 1997.

# A Appendix

## A.1 Language model training

The bidirectional LSTM language model was trained on the full set of protein domain sequences in the Pfam database, 21,827,419 total sequences. The language model was trained to predict the amino acid at position $i$ given observations of all amino acids before $i$ and all amino acids after $i$ by minimizing the cross entropy loss with log predicted log probabilities given by the sum of the forward and reverse LM direction predictions

$$\log p(x_i|x_{-i}) = \log p^F(x_i) + \log p^R(x_i)$$

where $p^F(x_i)$ is the probability given by the forward direction LSTM and $p^R(x_i)$ is the probability given by the reverse direction LSTM.

The language model architecture consisted of a 2-layer LSTM with 1024 units in each layer followed by a linear transformation into the 20-d amino acid prediction. All parameters were shared between the forward and reverse direction components. The model was trained for a single epoch using the Adam with a learning rate of 0.001 and minibatch size of 32.

## A.2 Structural similarity prediction benchmarks

For the NW-align method, similarity between protein sequences was computed using the BLOSUM62 substitution matrix with gap open and extend penalties of -11 and -1 respectively. For phmmer, each pair of sequences was compared in both directions (i.e. query->target and target->query) using the '–max' option. The similarity score for each pair was treated as the average off the query->target and target->query scores. For HHalign, multiple sequence alignments were first built for each sequence by using HHblits to search for similar sequences in the uniclust30 database with a maximum of 2 rounds of iteration (-n 2). Sequences pairs were then scored by using HHalign to score the target->query and query->target HMM-HMM alignments. Again, the average of the two scores was treated as the overall HHalign score. Finally, for TMalign, the structures for each pair of proteins were aligned and the scores for the query->target and target->query alignments were averaged to give the overall TMalign score for each pair of proteins.

To calculate the classification accuracy from the above scores, thresholds were found to maximize prediction accuracy when binning scores into similarity levels using 100,000 pairs of sequences sampled from the ASTRAL 2.06 training set.

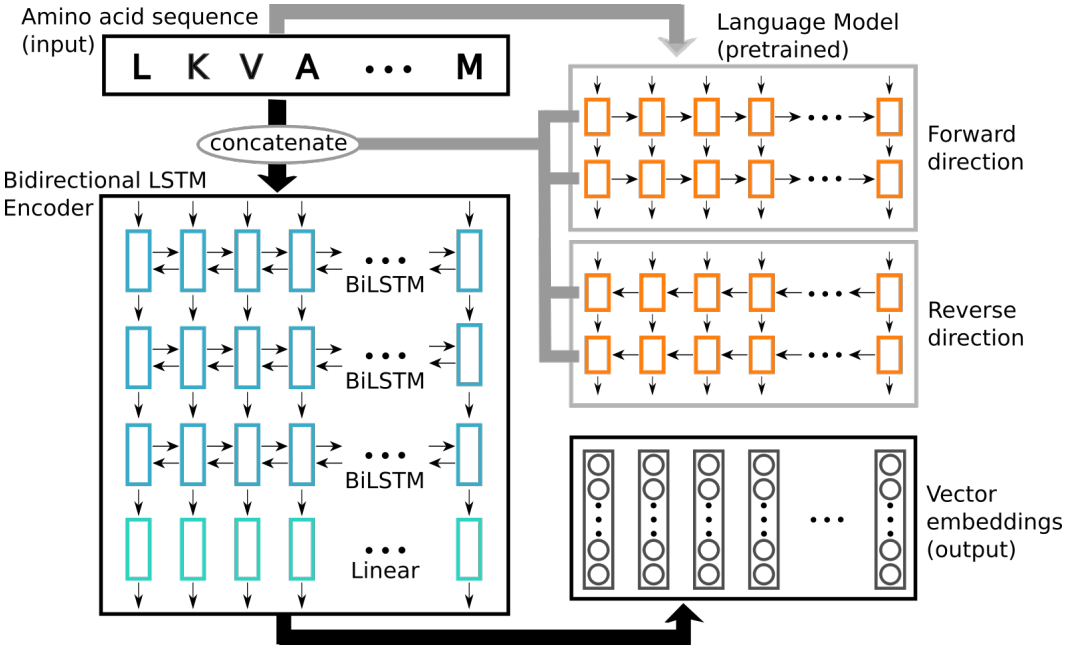


Figure 2: Illustration of the embedding model. The amino acid sequence is first past through the pretrained language model in both forward and reverse directions. The hidden states at each position of both directions of the language model are concatenated together with a one hot representation of the amino acids and past as input to the encoder. The final vector representations of each position of the amino acid sequence are given by a linear transformation of the outputs of the final bidirectional LSTM layer.

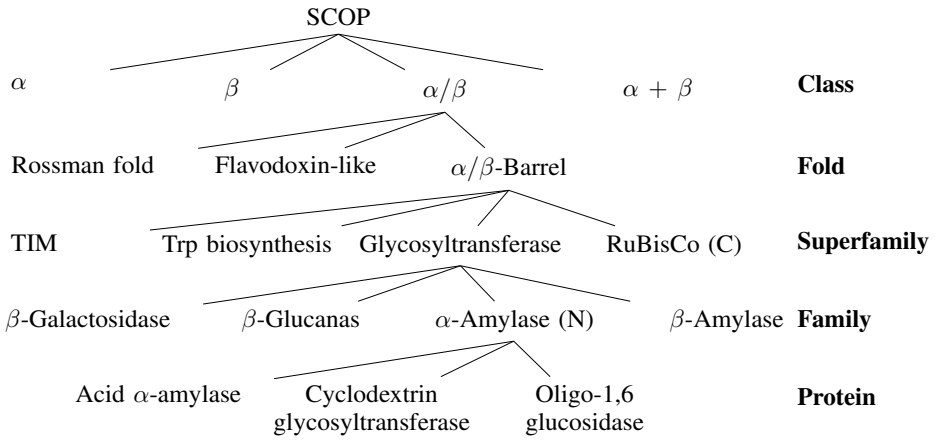


Figure 3: Illustration of the SCOP hierarchy modified from Hubbard et al. [44].

| **Embedding Model** | **Accuracy** | $r$ | $\rho$ |
|---|---|---|---|
| Linear | 0.85277 | 0.74419 | 0.60333 |
| Fully connected (1-layer, 512 units) | 0.91013 | 0.84193 | 0.67024 |
| BiLSTM (1-layer) | 0.92964 | 0.87239 | 0.67485 |
| BiLSTM (3-layer) | **0.93794** | **0.88048** | **0.67645** |

Table 4: Comparison of encoder architectures for the SSA model with LM inputs on the ASTRAL 2.06 test set.